AMENDMENTS

In the Claims

1. (Currently Amended) A method of determining an execution order for machine instructions to reduce spill code, said method comprising the step of:

from machine instructions that are ready for scheduling, scheduling the machine instruction for which an amount by which a size of a committed set of machine instructions would increase upon the scheduling of said machine instruction is smallest, said committed set of machine instructions including any machine instruction that is already scheduled and any machine instruction that is descendent from an already scheduled machine instruction, for each of said machine instructions ready for scheduling, said amount being determined by identifying descendent machine instructions of each of said machine instructions; and determining which of said descendent machine instructions and said machine instructions is not in said committed set of machine instructions; ~~and,~~

determining an execution order for the machine instructions to reduce spill code, a given machine instruction being considered ready for scheduling when scheduling of said given machine instruction as a next machine instruction would not cause an erroneous programmatic result; and,

undertaking the method when a risk of register overcommittedness exceeds a certain threshold, said threshold being exceeded when processor register availability drops below a particular threshold.

2 - 8. (Canceled)

9. (Currently Amended) A method of determining an execution order for machine instructions to reduce spill code, said method comprising the steps of:

in a first bit vector containing one bit to represent each machine instruction to be scheduled, setting those bits for which the represented machine instruction is not committed, and resetting the remaining bits;

for said each machine instruction to be scheduled that is ready for scheduling:

in a second bit vector also having one bit to represent each machine instruction to be ordered in the same sequence as in said first bit vector, setting those bits for which the represented machine instruction is a descendant of said each machine instruction that is ready for scheduling, and resetting the remaining bits;

performing a bitwise AND operation of said first bit vector and said second bit vector to create a third bit vector;

prior to performing said bitwise AND operation, setting in said second bit vector the bit for which the represented machine instruction is said each machine instruction that is ready for scheduling; and

determining the number of set bits in said third bit vector; and

selecting for execution the machine instruction for which said third bit vector contains a minimum number of set bits.

10 - 17. (Canceled)